
Subject: Re: interpolation problems in 2d and/or 3d space
Posted by [Paul Van Delst\[1\]](#) on Thu, 05 Jun 2008 13:19:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Baikal wrote:

> Hi Folks,
>
> One of funny and fundamental problems in interpolation business are as
> follows;
> In dealing with random walk problems such as oil spill particle
> tracking, we encounter randomly distributed particles on 2-D and 3-D
> space which are transpoted with ambient velocity defined by well-
> gridded and/or random grid (such as moving finite element node) so
> that I need to interpolate the velocities at current random points.
>
> I like to have a smooth flow field for these randomly moving
> particles. Does anybody have a smart idea for this type of
> interpolation?

I do 3-d interpolation for cloud and aerosol optical properties in atmospheric radiative transfer. Our model also computes the adjoint so the derivatives of the interpolation have to be continuous. What we settled upon doing was "effective 4-pt" polynomial interpolation - it's the weighted average of two quadratic polynomials that span and bookend the interval of interest.

It works well.

But I wrote the code in Fortran95. Sorry. The 1-D function is pretty simply:

```
! 1-D routine
SUBROUTINE Interp_1D(z, wlp, & ! Input
    z_int ) ! Output
! Arguments
REAL(fp),    INTENT(IN)    :: z(:)
TYPE(LPoly_type), INTENT(IN) :: wlp
REAL(fp),    INTENT(IN OUT) :: z_intreinitialisation
! Perform interpolation
z_int = ( wlp%w_left * ( wlp%lp_left(1) *z(1) + &
    wlp%lp_left(2) *z(2) + &
    wlp%lp_left(3) *z(3) ) ) + &
    ( wlp%w_right * ( wlp%lp_right(1)*z(2) + &
    wlp%lp_right(2)*z(3) + &
    wlp%lp_right(3)*z(4) ) )
END SUBROUTINE Interp_1D
```

where the weights (w_left and w_right) are simply the fractional distance from the nearest points.

cheers,

paulv
